EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	181	(application near2 move) with (network or node or grid or distribut$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:11
L2	2	(ghost near2 move) with (network or node or grid or distribut$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:11
S1	319	(CREAMER-THOMAS CREAMER-THOMAS-E CREAMER-THOMAS-EDWARD HILF-BILL-H HILF-B-H KATZ-NEIL KATZ-NEIL-A KATZ-NEIL-ALAN MOORE-VICTOR-S MOORE-VICTOR-STEWART MOORE-VICTOR-STUART).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/15 14:51
S2	10	("5592609" "5935006" "6024643" "20020002074" "6419577").did.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/15 14:50
S3	1832	714/38.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:12
S4	30	714/38.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:24
S5	5	714/38.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:24
S6	599	717/127.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 08:58
S7	11	717/127.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 08:57

EAST Search History

S8	5	717/127.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 08:58
S9	18	("6122664" "20020087949" "20020174415" "6587432" "6625648" "6671724" "6681243" "6714976" "20050149847").did.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 09:18
S10	10	("6122664" "6419577" "20040139202" "20060195559" "20050240777").did.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 09:20
S11	20	("6012152" "6083281" "6105059" "6266805" "6427000" "6430707" "20020162053" "20020174415" "20030014691" "20030028858").did.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 09:46
S12	26	("5515367" "5655081" "5802296" "5966441" "5974567" "6038399" "6108697" "6108703" "6229533" "6282697" "6389462" "6513159" "20040064548").did.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 13:10
S13	1859	714/25.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:25
S14	33	714/25.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:25
S15	3	714/25.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 08:52
S16	207	714/28.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:26
S17	7	714/28.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:26

EAST Search History

S18	1	714/28.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:27
S19	315	714/31.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:27
S20	9	714/31.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:27
S21	1	714/31.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:28
S22	483	714/33.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:27
S23	8	714/33.ccls. and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:27
S24	5	714/33.ccls. and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:28
S25	221	grid adj environment	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 12:44
S26	19	S25 and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 10:28
S27	13	(grid adj environment) and debug$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/16 12:56

EAST Search History

S28	82	714/25.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:31
S29	10	714/28.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:32
S30	25	714/31.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:40
S31	19	714/33.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:43
S32	119	714/38.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:52
S33	64	717/127.ccls. and @pd>="20061216"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 09:52
S34	1621	ghost and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 10:59
S35	607	ghost and grid and host and move	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 10:59
S36	37	ghost and grid and host and move and debug$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 11:04
S37	11	(ghost with debug$4) and grid	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:41

EAST Search History

S38	1194	grid adj (computing or environment)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:34
S39	97	S38 and debug$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:34
S40	21	(grid adj (computing or environment)) and ghost	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 11:24

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L4	597	714/38.ccls.	US-PGPUB	OR	ON	2007/05/30 13:25
L5	795	grid adj (computing or environment)	US-PGPUB	OR	ON	2007/05/30 13:34
L6	87	L5 and debug$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/30 13:34
L7	10	(ghost with debug$4) and grid	US-PGPUB	OR	ON	2007/05/30 13:41

Debugging a grid environment using **ghost** agents - Patent 20050066310
A method for **debugging** software objects within a **grid** environment including the step identifying a host, wherein the host is a software object. A **ghost** ...
www.freepatentsonline.com/20050066310.html - 66k - Cached - Similar pages

Validating software in a **grid** environment using **ghost** agents ...
A **ghost** agent can be associated with the host, where the **ghost** agent ca. ... Next Patent (**Debugging** a **grid** environment using ...) -> ...
www.freepatentsonline.com/20050066309.html - 73k - Cached - Similar pages
[More results from www.freepatentsonline.com]

HPC @ IU - Putting TotalView to use: **Debugging** parallelDiffusion
Also set my **ghost** points. These are **grid** points not owned by */ /* me but ... you a good idea of how you could use TotalView to **debug** your parallel code! ...
rc.uits.iu.edu/hpc/mpi_tutorial_2005/s3_tv_paralleldiffusion.html - 16k - Cached - Similar pages

How to use Ghostscript
Setting this to 2 enables automatic **grid** fitting for True Type glyphs. ..... In the table below, the first column is a **debugging** switch, the second is an ...
www.cs.wisc.edu/~**ghost**/doc/cvs/Use.htm - 149k - Cached - Similar pages

[Paper] Transparent Migration of Multi-Threaded Applications on a ...
Shell is attached to the pro- grammer provided **Ghost** when entity is instantiated. ... Beginning with Java 1.4, JVM runs at full speed in **debug** mode[21] Exe- ...
www.actapress.com/PDFViewer.aspx?paperId=27510 - Similar pages

I'm still living with your **ghost**...
"I'm still living with your **ghost**" Santa Monica - Everclear ... How long has it been since I've had to **debug** 3rd a 3rd party **grid** library? Never, maybe. ...
geekswithblogs.net/yasko/archive/2007/04/11/111404.aspx - 19k - Cached - Similar pages

[PDF] TRANSPARENT MIGRATION OF MULTI-THREADED APPLICATIONS ON A JAVA ...
File Format: PDF/Adobe Acrobat - View as HTML
Ghost Ghost represents the programmer provided part of. an entity. **Ghost** implements the actual ... Java 1.4, JVM runs at full speed in **debug** mode[21] Exe- ...
arxiv.org/pdf/cs/0608116 - Similar pages

[PDF] A Test Suite for High-Performance Parallel Java
File Format: PDF/Adobe Acrobat - View as HTML
easier to create, **debug**, and maintain. The elegant threading provides a simple route to ..... Internal **Grid** point. Boundary **Grid** point. **Ghost Grid** point ...
cacr.library.caltech.edu/25/01/cacr178.pdf - Similar pages

2: static char help[] ="Solves a time-dependent nonlinear PDE ...
Runtime options include:\n\ 4: -M <xg>, where <xg> = number of **grid** points\n\ 5: -**debug** : Activate **debugging** printouts\n\ 6: -nox : Deactivate x-window ...
www-unix.mcs.anl.gov/petsc/petsc-2/snapshots/petsc-current/src/ts/examples/tutorials/ex2.c.html - 65k - Cached - Similar pages

7 Multiblock Framework API Reference
Print a **debugging** representation of the framework's information about the current ... varies, intent(inout) :: **grid**. Update the values in the **ghost** regions ...
charm.cs.uiuc.edu/manuals/html/mblock/7Multiblock_Framework_API_R.html - 35k -
Cached - Similar pages

1 2 3 4 5 6 7 8 9 10 **Next**

Terms used **ghost** AND **grid** AND **debug** Found **1,152** of **201,890**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Level set and PDE methods for computer graphics
David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(17.07 MB) Additional Information: full citation, abstract, citings

Level set methods, an important class of partial differential equation (PDE) methods, define dynamic surfaces implicitly as the level set (iso-surface) of a sampled, evolving nD function. The course begins with preparatory material that introduces the concept of using partial differential equations to solve problems in computer graphics, geometric modeling and computer vision. This will include the structure and behavior of several different types of differential equations, e.g. the level set eq ...

2 The elements of nature: interactive and realistic techniques
Oliver Deusen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(17.65 MB) Additional Information: full citation, abstract

This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a nice balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

3 Verification: Modeling wildcard-free MPI programs for verification
Stephen F. Siegel, George S. Avrunin
June 2005 **Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '05**
Publisher: ACM Press
Full text available: pdf(292.09 KB) Additional Information: full citation, abstract, references, citings, index terms

We give several theorems that can be used to substantially reduce the state space that must be considered in applying finite-state verification techniques, such as model checking, to parallel programs written using a subset of MPI. We illustrate the utility of

these theorems by applying them to a small but realistic example.

Keywords: MPI, SPIN, analysis, concurrent systems, deadlock, finite-state verification, formal methods, message passing interface, model checking, parallel computation

4 GPGPU: general purpose computation on graphics hardware

David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn

August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**

Publisher: ACM Press

Full text available: pdf(63.03 MB) Additional Information: full citation, abstract, citings

The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

5 OOPSLA onward! track chair's welcome: Collaborative diffusion: programming antiobjects

Alexander Repenning

October 2006 **Companion to the 21st ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**

Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, abstract, references, index terms

Object-oriented programming has worked quite well - so far. What are the objects, how do they relate to each other? Once we clarified these questions we typically feel confident to design and implement even the most complex systems. However, objects can deceive us. They can lure us into a false sense of understanding. The metaphor of objects can go too far by making us try to create objects that are too much inspired by the real world. This is a serious problem, as a resulting system may be sign ...

Keywords: collaborative agents, diffusion, distributed artificial intelligence, end-user programming, game AI, incremental AI, multi-agent architecture, object-oriented programming, psychology of programming

6 Performance analysis and optimization on the UCLA parallel atmospheric general circulation model code

John Lou, John Farrara

November 1996 **Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '96**

Publisher: IEEE Computer Society

Full text available: pdf(126.49 KB) Additional Information: full citation, abstract, references, citings, index terms

An analysis is presented of several factors influencing the performance of a parallel implementation of the UCLA atmospheric general circulation model(AGCM) on massively parallel computer systems. Several modifications to the parallel AGCM code aimed at improving its numerical efficiency, interprocessor communication cost, load-balance and cache efficiency are discussed. The impact of some of the optimization strategies on the performance of the AGCM code as we implemented on several state- ...

7 Excuse me, I need better AI!: employing collaborative diffusion to make game AI

child's play
Alexander Repenning
July 2006 **Proceedings of the 2006 ACM SIGGRAPH symposium on Videogames sandbox '06**
Publisher: ACM Press
Full text available: pdf(547.39 KB) Additional Information: full citation, abstract, references, index terms

The idea of end-user game authoring environments is quickly gaining momentum in education. Environments such as AgentSheets have been used by thousands of children to learn about programming and design by creating their own computer games. With only hours of training these children initially create their own versions of classical games such as Frogger, Sokoban, and Space Invaders and later begin to design and implement their own game ideas. After creating numerous simple games including cursor c ...

Keywords: collaborative agents, diffusion, distributed artificial intelligence, end-user programming, game AI, incremental AI, multi-agent architecture, object-oriented programming, psychology of programming

8 Co-array Fortran for parallel programming
Robert W. Numrich, John Reid
August 1998 **ACM SIGPLAN Fortran Forum**, Volume 17 Issue 2
Publisher: ACM Press
Full text available: pdf(1.94 MB) Additional Information: full citation, abstract, citings, index terms

Co-Array Fortran, formerly known as F--, is a small extension of Fortran 95 for parallel processing. A Co-Array Fortran program is interpreted as if it were replicated a number of times and all copies were executed asynchronously. Each copy has its own set of data objects and is termed an image. The array syntax of Fortran 95 is extended with additional trailing subscripts in square brackets to give a clear and straightforward representation of any access to data that is spread across ...

9 New life in dusty decks: results of porting a CM Fortran based aeroacoustic model to high performance Fortran
Jeffrey J. Nucciarone, Yusuf Özyörük, Lyle N. Long
November 1997 **Proceedings of the 1997 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '97**
Publisher: ACM Press
Full text available: pdf(110.48 KB) Additional Information: full citation, abstract, references

The High Performance Fortran language is a 'standard by consensus', developed by individuals and vendors in the high performance computing industry, to provide a low barrier entry to parallel computing. It promises to be an easier to use development environment for distributed memory computing platforms compared to the programming complexity required by message passing libraries such as PVM and MPI. HPF promises much and is still in its infancy.Since HPF was developed in part based on experience ...

Keywords: Thinking Machines CM Fortran, aeroacoustics, dusty decks, high performance Fortran

10 The design and development of ZPL
Lawrence Snyder
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III**
Publisher: ACM Press
Full text available: Additional Information:

pdf(2.65 MB) full citation, abstract, references, index terms

ZPL is an implicitly parallel programming language, which means all instructions to implement and manage the parallelism are inserted by the compiler. It is the first implicitly parallel language to achieve performance portability, that is, consistent high performance across all (MIMD) parallel platforms. ZPL has been designed from first principles, and is founded on the CTA abstract parallel machine. A key enabler of ZPL's performance portability is its What You See Is What You Get (WYSIWYG) ...

Keywords: CTA, WYSIWYG performance model, parallel language design, performance portability, problem space promotion, regions, type architecture

11 Evaluating titanium SPMD programs on the Tera MTA

Carleton Miyamoto, Chang Lin
January 1999 **Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM) Supercomputing '99**
Publisher: ACM Press
Full text available: pdf(314.08 KB) Additional Information: full citation, references, index terms

12 Multithreaded, multicore, and SoC systems: The potential of the cell processor for scientific computing

Samuel Williams, John Shalf, Leonid Oliker, Shoaib Kamil, Parry Husbands, Katherine Yelick
May 2006 **Proceedings of the 3rd conference on Computing frontiers CF '06**
Publisher: ACM Press
Full text available: pdf(285.38 KB) Additional Information: full citation, abstract, references, index terms

The slowing pace of commodity microprocessor performance improvements combined with ever-increasing chip power demands has become of utmost concern to computational scientists. As a result, the high performance computing community is examining alternative architectures that address the limitations of modern cache-based designs. In this work, we examine the potential of using the forthcoming STI Cell processor as a building block for future high-end computing systems. Our work contains several no ...

Keywords: FFT, GEMM, SpMV, cell processor, sparse matrix, stencil, three level memory

13 GPGPU: general-purpose computation on graphics hardware: A streaming narrow-band algorithm: interactive computation and visualization of level sets

Aaron E. Lefohn, Joe M. Kniss, Charles D. Hansen, Ross T. Whitaker
July 2005 **ACM SIGGRAPH 2005 Courses SIGGRAPH '05**
Publisher: ACM Press
Full text available: pdf(405.14 KB) Additional Information: full citation, abstract, references

Deformable isosurfaces, implemented with level-set methods, have demonstrated a great potential in visualization and computer graphics for applications such as segmentation, surface processing, and physically-based modeling. Their usefulness has been limited, however, by their high computational cost and reliance on signi£cant parameter tuning. This paper presents a solution to these challenges by describing graphics processor (GPU) based algorithms for solving and visualizing level-set so ...

Keywords: GPU, deformable models, image segmentation, level sets, streaming computation, virtual memory, volume visualization

14 The performance realities of massively parallel processors: a case study

O. M. Lubeck, M. L. Simmons, H. J. Wasserman

December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing Supercomputing '92**

Publisher: IEEE Computer Society Press

Full text available: pdf(874.40 KB) Additional Information: full citation, references, citings, index terms

15 Tools: DART: a toolkit for rapid design exploration of augmented reality experiences

Blair MacIntyre, Maribeth Gandy, Steven Dow, Jay David Bolter

October 2004 **Proceedings of the 17th annual ACM symposium on User interface software and technology UIST '04**

Publisher: ACM Press

Full text available: pdf(2.94 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we describe The Designer's Augmented Reality Toolkit (DART). DART is built on top of Macromedia Director, a widely used multimedia development environment. We summarize the most significant problems faced by designers working with AR in the real world, and discuss how DART addresses them. Most of DART is implemented in an interpreted scripting language, and can be modified by designers to suit their needs. Our work focuses on supporting early design activities, especially a rap ...

Keywords: animatics, augmented reality, capture/replay, design environments, mixed reality, storyboards

16 Simple, state-based approaches to program-based anomaly detection

C. C. Michael, Anup Ghosh

August 2002 **ACM Transactions on Information and System Security (TISSEC)**, Volume 5 Issue 3

Publisher: ACM Press

Full text available: pdf(459.57 KB) Additional Information: full citation, abstract, references, citings, index terms

This article describes variants of two state-based intrusion detection algorithms from Michael and Ghosh [2000] and Ghosh et al. [2000], and gives experimental results on their performance. The algorithms detect anomalies in execution audit data. One is a simply constructed finite-state machine, and the other two monitor statistical deviations from normal program behavior. The performance of these algorithms is evaluated as a function of the amount of available training data, and they are compar ...

Keywords: Anomaly detection, finite automata, information system security, intrusion detection, machine learning

17 Cursor control and pointing devices: The migratory cursor: accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations

Yoshiyuki Mihara, Etsuya Shibayama, Shin Takahashi

October 2005 **Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility Assets '05**

Publisher: ACM Press

Full text available: pdf(893.07 KB) mov(18:43 MIN) Additional Information: full citation, abstract, references, index terms

We present the migratory cursor, which is an interactive interface that enables users to move a cursor to any desired position quickly and accurately using voice alone. The

migratory cursor combines discrete specification that allows a user to specify a location quickly, but approximately, with continuous specification that allows the user to specify a location more precisely, but slowly. The migratory cursor displays multiple ghost cursors that are aligned vertically or horizontally with the ac ...

Keywords: non-verbal voice input, speech-based cursor movement

18 Applications: Near-optimal adaptive control of a large grid application

Det Buaklee, Gregory F. Tracy, Mary K. Vernon, Stephen J. Wright

June 2002 **Proceedings of the 16th international conference on Supercomputing ICS '02**

Publisher: ACM Press

Full text available: pdf(251.23 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper develops a performance model that is used to control the adaptive execution the ATR code for solving large stochastic optimization problems on computational grids. A detailed analysis of the execution characteristics of ATR is used to construct the performance model that is then used to specify (a) near-optimal dynamic values of parameters that govern the distribution of work, and (b) a new task scheduling algorithm. Together, these new features minimize ATR execution time on any coll ...

Keywords: adaptive computations, grid computing, parallel algorithms, parallel application performance, stochastic optimization

19 Technical papers: Grid applications---Parallel genomic sequence-searching on an ad-hoc grid: experiences, lessons learned, and implications

Mark K. Gardner, Wu-chun Feng, Jeremy Archuleta, Heshan Lin, Xiaosong Mal

November 2006 **Proceedings of the 2006 ACM/IEEE conference on Supercomputing SC '06**

Publisher: ACM Press

Full text available: pdf(658.37 KB) html(2.36 KB) Additional Information: full citation, abstract, references

The Basic Local Alignment Search Tool (BLAST) allows bioinformaticists to characterize an unknown sequence by comparing it against a database of known sequences. The similarity between sequences enables biologists to detect evolutionary relationships and infer biological properties of the unknown sequence.mpiBLAST, our parallel BLAST, decreases the search time of a 300 KB query on the current NT database from over two full days to under 10 minutes on a 128-processor cluster and allows larger que ...

Keywords: BLAST, agile development, bioinformatics, cluster computing, fault tolerance, grid computing, optical networking, scalability, scheduling, scripting, sequence search

20 Pursuing scalability for *hypre*'s conceptual interfaces

Robert D. Falgout, Jim E. Jones, Ulrike Meier Yang

September 2005 **ACM Transactions on Mathematical Software (TOMS)**, Volume 31 Issue 3

Publisher: ACM Press

Full text available: pdf(564.71 KB) Additional Information: full citation, abstract, references, citings, index terms

The software library *hypre* provides high-performance preconditioners and solvers for the solution of large, sparse linear systems on massively parallel computers as well as conceptual interfaces that allow users to access the library in the way they naturally think

about their problems. These interfaces include a stencil-based structured interface (Struct); a semistructured interface (semiStruct), which is appropriate for applications that are mostly structured, for example, block structu ...

Keywords: User interfaces, parallel programming, scalability

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next